2017-02

Convex optimization for proximity maneuvering of a spacecraft with a robotic manipulator

Virgili-Llop, Josep

To appear in the proceedings of the 27th AAS/AIAA Spaceflight Mechanics Meeting, San Antonio, TX
February 6-9, 2017

AAS 17-418

CONVEX OPTIMIZATION FOR PROXIMITY MANEUVERING OF A SPACECRAFT WITH A ROBOTIC MANIPULATOR

Josep Virgili-Llop[*], Costantinos Zagaris[†], Richard Zappulla II [†], Andrew Bradstreet [‡], and Marcello Romano [§]

This paper presents a convex optimization-based guidance algorithm for maneuvering a spacecraft equipped with a robotic manipulator. A local solution to the original optimization problem is found by solving a collection of simpler convex programming problems. Given the deterministic convergence properties of convex programming, the proposed algorithm can be implemented onboard a spacecraft and used for real-time applications. To reduce the complexity of the original optimization problem, we first divide the maneuver into two simultaneously occurring sub-maneuvers: a system-wide translation and an internal re-configuration. These two sub-maneuvers are individually optimized using a sequential convex optimization approach to overcome the presence of non-convex inequality and nonlinear equality constraints. The paradigmatic example of capturing a tumbling object is used throughout the paper to illustrate the use of the proposed optimization approach. Additionally, a new explicitly convex formulation of a line-of-sight constraint is introduced.

INTRODUCTION

Capturing a tumbling Resident Space Object (RSO), with a chaser spacecraft equipped with a robotic manipulator, is a required maneuver for many future space missions.[1] The highly nonlinear dynamics of spacecraft-manipulator systems and the presence of nonlinear constraints make the optimization of the capture maneuver a challenging task. Traditional nonlinear optimization approaches can yield optimal solutions, but their significant computational cost renders them unfit for real-time onboard optimization.[2–7] As the future rotational state of a tumbling RSO can be difficult to predict ahead of time,[8] real-time onboard optimization is the only viable alternative for a wide range of capture targets.

To obtain an optimization approach suitable for onboard implementation, we propose to divide the complex capture maneuver into two simpler sub-maneuvers. These two sub-maneuvers are then individually optimized using a sequential convex optimization approach. The first optimization problem solves the system-wide translation, exploiting the simpler whole-body translational dynamics. The second optimization problem solves the system's internal re-configuration. By dividing the maneuver we simplify the optimization, but also make certain assumptions that inevitably degrade the optimality of the obtained solution.

[*]Research Associate, Spacecraft Robotics Laboratory, Naval Postgraduate School, 1 University Circle, CA, 93940 USA
[†]PhD Candidate, Spacecraft Robotics Laboratory, Naval Postgraduate School
[‡]Graduate Student, Spacecraft Robotics Laboratory, Naval Postgraduate School
[§]Associate Professor, Spacecraft Robotics Laboratory, Naval Postgraduate School

The first optimization step, solving the system-wide translation, produces the control history (chaser forces) required to translate the chaser's center-of-mass to a location where the RSO's grappling fixture is within the chaser's manipulator reach. A sequential convex optimization approach is used to circumvent the presence of the non-convex state inequality constraints introduced by collision avoidance keep-out zones.[9] As the internal state of the system (chaser orientation and manipulator configuration) is undefined during this first optimization step, the keep-out zone constraints are conservatively applied to a sphere that fully contains all possible spacecraft-manipulator configurations.

The second optimization step, solving the internal re-configuration, produces the control history (manipulator and base-spacecraft torques) that re-configures the manipulator and re-orients the chaser with minimum control effort. In this case, a sequential convex optimization approach is used to circumvent the nonlinear equality constraints emerging from the nonlinear manipulator dynamics and differential quaternion kinematics.[10] In this second optimization step we introduce an explicitly convex line-of-sight constraint formulation.

The approach presented in this paper obtains a local solution to the original optimal control problem by solving a collection of convex programming problems. Modern algorithms used to solve convex programming problems, such as interior-point methods,[11] are computationally efficient and enjoy guaranteed convergence in polynomial time. These properties have been exploited to solve optimal control problems on embedded systems and in real-time.[12,13]

The approach to convexify the optimal control problem of an RSO capture maneuver has been inspired by the remarkable success obtained by convex optimization in the areas of "powered soft-landing"[14] and "rendezvous and proximity operations".[15]

In this paper, the approach and implications of dividing the maneuver into two sub-maneuvers are discussed first. Then, both sub-maneuver optimization steps are discussed in detail. Finally, the validity of the proposed approach is verified by numerical simulations using a realistic RSO capture scenario.

PROBLEM FORMULATION AND APPROACH

Figure 1 notionally shows the initial configuration of the chaser and the tumbling RSO to be captured. When formulating the problem, the following underlying assumptions have been made:

- Both the chaser and the target RSO are composed of rigid bodies.
- Environmental forces (gravity gradient, solar radiation pressure, *etc.*) as well as the relative orbital dynamic effects are negligible. This can be justified by the short duration of the maneuver and the close proximity of the two vehicles.
- The state and dynamic properties of the chaser and of the target RSO are observable or known a priori.
- The desired manipulator pose to be reached at grasping (*i.e.*, end of the capture maneuver) is pre-set $\boldsymbol{\theta}_f$. This final desired configuration can be notionally seen in Figure 2.

The optimization of the capture maneuver can be simplified if the whole maneuver is conceptually divided into two simultaneously occurring sub-maneuvers. One sub-maneuver being the system-wide translation of the combined multibody system center-of-mass, and the other sub-maneuver

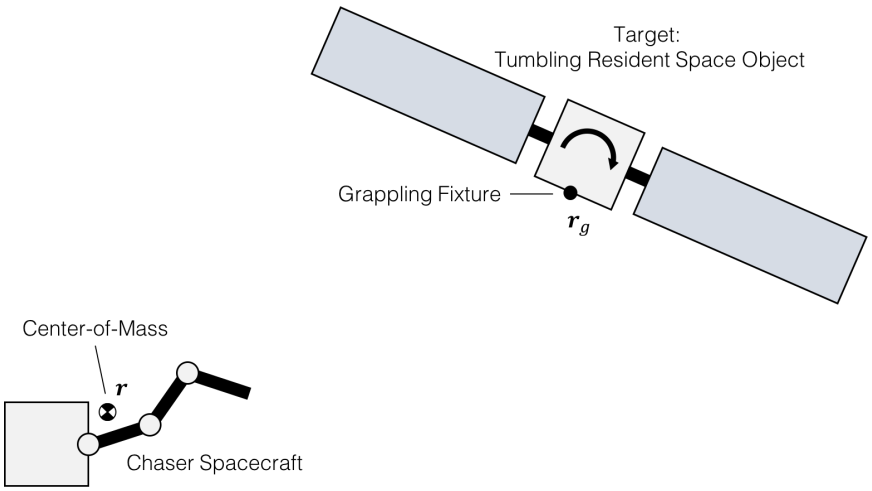


Figure 1. Notional Representation of the Chaser and Tumbling RSO to be Captured.

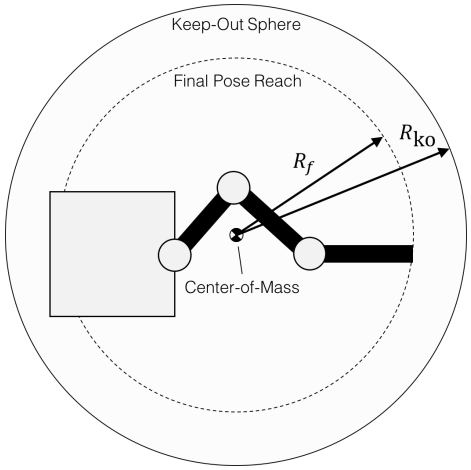


Figure 2. Notional Representation of the Manipulator's Grasping Pose, Final Pose Reach R_f, and Chaser's Keep-Out Sphere R_{ko}.

being the re-configuration of the multibody system around its center-of-mass. Although these two sub-maneuvers are linked through the system's dynamics, they can be individually optimized if certain simplifying assumptions are made.

In the proposed approach, the system-wide translation sub-maneuver is optimized first (Step 1). The results of this first optimization are later used to optimize the internal re-configuration sub-maneuver (Step 2).

Cleanly dividing the optimization into two sequential optimization steps has important implications on the resulting solution. Briefly stated, optimizing the system-wide translation first, prioritizes the reduction of the system-wide translation cost, even at the expense of a higher internal re-configuration cost. This prioritization can be justified by comparing the costs associated with the two sub-maneuvers. Translational maneuvers must make use of the spacecraft's limited supply of propellant, while internal re-configuration maneuvers can make use of momentum exchange devices. The cost associated with the use of propellant is usually higher than the cost associated with

the electrical power required to drive the momentum exchange devices and manipulator joints.

Another consequence of optimizing the system-wide translation sub-maneuver first is that the internal configuration of the spacecraft-manipulator system is undefined during this first optimization step. Therefore, a conservative stance must be taken to guarantee a collision-less maneuver. In this case, it will be conservatively assumed that the spacecraft-manipulator system is confined within a sphere of a given radius R_{ko} (see Figure 2). The keep-out zone constraints are then applied to this enclosing sphere, ensuring a collision-free maneuver.

These assumptions unavoidably produce a conservative and sub-optimal solution. However, we argue that the reduced computational complexity of the resulting optimization procedure outweighs the loss of global optimality.

STEP 1: SYSTEM-WIDE TRANSLATION OPTIMIZATION

The motion of the chaser's center-of-mass $\boldsymbol{r}$ during a system-wide translation is a function of the total system mass m and of the forces $\boldsymbol{F}$ applied to it. The equation governing the motion of the chaser's center-of-mass can be simply written as follows:

$$\boldsymbol{F} = m\ddot{\boldsymbol{r}} \tag{1}$$

As the capture maneuver is expected to be short (in the order of minutes) and the required Δv small, it can be assumed that the chaser mass will remain constant. Under this assumption, it is appropriate to minimize the L_1-norm of the force $\boldsymbol{F}$, using the following cost function:

$$J = \int_0^{t_f} \|\boldsymbol{F}(t)\|_1 \, dt \tag{2}$$

Using this cost function generates minimum propellant maneuvers with a bang-off-bang control style. That means that the chaser will abruptly decelerate just before capturing its target.

A more gentle translation maneuver can be obtained by using a quadratic cost function, with $\boldsymbol{W_F}$ denoting a positive definite weight matrix.

$$J = \int_0^{t_f} \boldsymbol{F}^T \boldsymbol{W_F} \boldsymbol{F} dt \tag{3}$$

In addition to ensure a less aggressive final approach, a quadratic cost function leads to numerically more stable solutions (as solutions do not have discontinuities). For these two reasons we use a quadratic cost function of the type shown in Eq. (3) in our formulation.

The amount of force that the chaser thrusters can produce is bounded, thus imposing the following constraint:

$$\|\boldsymbol{F}(t)\|_2 \leq F_{\max} \tag{4}$$

As the manipulator's grasping configuration is pre-set, the final pose reach R_f, as shown in Figure 2, is known. This final pose reach can be used to constrain the terminal position of the chaser's center-of-mass $\boldsymbol{r}(t_f)$. In this final position, the grappling fixture $\boldsymbol{r}_g(t_f)$ has to be within grasping distance R_f. This terminal condition can be formally expressed as follows:

$$\|\boldsymbol{r}_g(t_f) - \boldsymbol{r}(t_f)\|_2 \leq R_f \tag{5}$$

The terminal velocity of the chaser is constrained to achieve a zero relative velocity. The RSO's angular velocity is denoted by $\boldsymbol{\omega}_{\mathrm{RSO}}$ and the position of its center-of mass by $\boldsymbol{r}_{\mathrm{RSO}}$.

$$\dot{\boldsymbol{r}}(t_f) = \boldsymbol{\omega}_{\mathrm{RSO}}(t_f) \times (\boldsymbol{r}(t_f) - \boldsymbol{r}_{\mathrm{RSO}}(t_f)) \tag{6}$$

In order to avoid collisions of the chaser with the RSO, keep-out zone constraints are introduced. It is assumed that the geometry of the RSO is known and that its state can be predicted for the short duration of the maneuver. Unfortunately, the internal configuration of the chaser is undefined during this first optimization step. To guarantee a collision-free trajectory, these keep-out zone constraints can be applied to a sphere of radius R_{ko}, as shown in Figure 2, that fully contains the chaser system (regardless of the manipulator configuration or chaser's orientation). In general, the keep-out sphere will be coincident with the "floating-base reachable workspace".[16]

If the chaser sphere is denoted by S_{chaser} and the P keep-out zones of the RSO by S_{i} (for $i = 1, \ldots, P$), then the keep-out zone constraints can be formally expressed using the following expression:

$$S_{\mathrm{chaser}}(t) \cap S_i(t) = \emptyset \text{ for } i = 1, \ldots, P \tag{7}$$

The resulting optimization problem, using the quadratic cost function introduced in Eq. (3), can be formulated as follows:

$$\text{minimize: } \int_0^{t_f} \boldsymbol{F}^T \boldsymbol{W}_{\boldsymbol{F}} \boldsymbol{F} dt \tag{8a}$$

$$\text{subject to: } \boldsymbol{F} = m\ddot{\boldsymbol{r}} \tag{8b}$$

$$\|\boldsymbol{F}(t)\|_2 \leq F_{\max} \tag{8c}$$

$$S_{\mathrm{chaser}}(t) \cap S_i(t) = \emptyset \text{ for } i = 1, \ldots, P \tag{8d}$$

$$\|\boldsymbol{r}_g(t_f) - \boldsymbol{r}(t_f)\|_2 \leq R_f \tag{8e}$$

$$\dot{\boldsymbol{r}}(t_f) = \boldsymbol{\omega}_{\mathrm{RSO}}(t_f) \times (\boldsymbol{r}(t_f) - \boldsymbol{r}_{\mathrm{RSO}}(t_f)) \tag{8f}$$

The keep-out zone constraints (see Eq. (8d)) can be re-formulated using the signed distance d_i (see Figure 3) between S_{chaser} and S_i (analogous to the Minkowski difference between the two sets).[9]

$$d_i \geq 0 \text{ for } i = 1, \ldots, P \tag{9}$$

This signed distance d_i, which is positive when the two sets are not in contact, and negative when they intersect each other, can be formally defined as follows:

$$d_i = d_i^+ - d_i^- \tag{10a}$$

$$d_i^+ = \inf\{\|\boldsymbol{d}\|_2 \mid (S_{\mathrm{chaser}} + \boldsymbol{d}) \cap S_i \neq \emptyset\} \tag{10b}$$

$$d_i^- = \inf\{\|\boldsymbol{d}\|_2 \mid (S_{\mathrm{chaser}} + \boldsymbol{d}) \cap S_i = \emptyset\} \tag{10c}$$

An efficient algorithm to compute the d_i^+ distance is the Gilbert-Johnson-Keerthi algorithm,[17] while the Expanding Polytope Algorithm[18] can be used to efficiently compute the d_i^- distance.

The keep-out zone constraints in Eq. (9) are, in general, non-convex. The presence of non-convex inequality constraints prevents us from directly casting this problem using a convex optimization formulation. To circumvent the presence of these non-convex constraints, a sequential convex optimization approach is used.

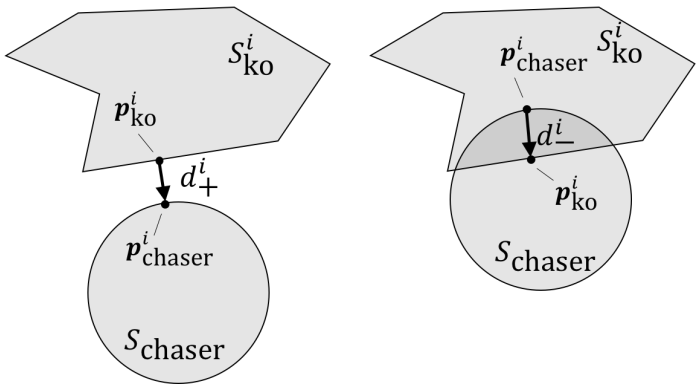


Figure 3. Notional Representation of the Signed Distances d_i^+ and d_i^-.

In a sequential convex optimization approach, the original non-convex problem is solved by repeatedly solving a convex approximation of the original problem around the previous iteration solution ${}^{[k-1]}\boldsymbol{F}^\star$. A locally optimal solution to the original problem is obtained when the solutions of two consecutive iterations converge below a certain threshold ϵ.

$$\left\| {}^{[k-1]}\boldsymbol{F}^\star - {}^{[k]}\boldsymbol{F}^\star \right\|_1 \leq \epsilon \tag{11}$$

By linearizing the keep-out zone constraints around a given trajectory $\tilde{\boldsymbol{r}}$, derived from the previous solution ${}^{[k-1]}\boldsymbol{F}^\star \rightarrow \tilde{\boldsymbol{r}}$, a convex approximation of the signed distance d_i (shown in Eq. (10a)) is obtained.[9]

$$d_i \approx \tilde{d}_i + \hat{\boldsymbol{d}}_i^T \boldsymbol{J}_i \left(\boldsymbol{r} - \tilde{\boldsymbol{r}}\right) \tag{12a}$$

$$\boldsymbol{d}_i = \begin{cases} \boldsymbol{p}_i^{\text{chaser}} - \boldsymbol{p}_i^{\text{ko}}, \text{for } d_i > 0 \\ \boldsymbol{p}_i^{\text{ko}} - \boldsymbol{p}_i^{\text{chaser}}, \text{for } d_i < 0 \end{cases} \tag{12b}$$

$$\hat{\boldsymbol{d}}_i = \frac{\tilde{\boldsymbol{d}}_i}{\left\| \tilde{\boldsymbol{d}}_i \right\|_2} \tag{12c}$$

The $\boldsymbol{p}_i^{\text{ko}} \in S_i$ and $\boldsymbol{p}_i^{\text{chaser}} \in S_{\text{chaser}}$ are the closest supporting points as shown in Figure 3 and the $\boldsymbol{J}_i$ matrix denotes the analytical Jacobian of $\boldsymbol{p}_i^{\text{chaser}}$. As S_{chaser} is a sphere, $\boldsymbol{J}_i$ is the identity matrix.

Ultimately, the non-convex keep-out zone constraints in Eq. (9) can be approximated by the following convex constraint:

$$\tilde{d}_i + \hat{\boldsymbol{d}}_i^T \left(\boldsymbol{r} - \tilde{\boldsymbol{r}}\right) \geq 0 \text{ for } i = 1, \ldots, P \tag{13}$$

An important element of sequential convex optimization is to constrain the solution to remain within a trust region where the convex approximations are still valid. This can be enforced by adding the following additional constraint, where $\rho_{\boldsymbol{r}}$ denotes the extent of this trust region:

$$\left\| \boldsymbol{r} - \tilde{\boldsymbol{r}} \right\|_1 \leq \rho_{\boldsymbol{r}} \tag{14}$$

Finally, the problem is discretized using a direct transcription method with N nodes. For the problem to remain convex, it must be formulated as a fixed-time problem. A method to optimize the final time t_f using an outer loop optimization is discussed elsewhere.[19]

The resulting fixed-time convex optimization problem, to be iteratively solved, can be written as follows:

$$\text{minimize: } \sum_{n=1}^{N} {\boldsymbol{F}^{[n]}}^T \boldsymbol{W}_{\boldsymbol{F}} \boldsymbol{F}^{[n]} \Delta t^{[n]} \tag{15a}$$

$$\text{subject to: } \begin{bmatrix} \boldsymbol{r}^{[n+1]} \\ \dot{\boldsymbol{r}}^{[n+1]} \end{bmatrix} = \boldsymbol{\Phi}_{\boldsymbol{r}}^{[n]} \begin{bmatrix} \boldsymbol{r}^{[n]} \\ \dot{\boldsymbol{r}}^{[n]} \end{bmatrix} + \boldsymbol{\Psi}_{\boldsymbol{r}}^{[n]} \frac{\boldsymbol{F}^{[n]}}{m} \qquad \text{for } n = 2, \ldots, N \tag{15b}$$

$$\left\| \boldsymbol{F}^{[n]} \right\|_2 \leq F_{\max} \qquad \text{for } n = 1, \ldots, N \tag{15c}$$

$$\tilde{d}_i^{[n]} + \hat{\boldsymbol{d}}_i^{[n]} \left(\boldsymbol{r}^{[n]} - \tilde{\boldsymbol{r}}^{[n]} \right) \geq 0 \qquad \text{for } i = 1, \ldots, P \text{ and } n = 1, \ldots, N \tag{15d}$$

$$\left\| \boldsymbol{r}_g^{[n]} - \boldsymbol{r}^{[n]} \right\|_2 \leq R_f \qquad \text{for } n = N \tag{15e}$$

$$\dot{\boldsymbol{r}}^{[n]} = \boldsymbol{\omega}_{\text{RSO}}^{[n]} \times \left(\boldsymbol{r}^{[n]} - \boldsymbol{r}_{\text{RSO}}^{[n]} \right) \qquad \text{for } n = N \tag{15f}$$

$$\left\| \boldsymbol{r}^{[n]} - \tilde{\boldsymbol{r}}^{[n]} \right\|_1 \leq \rho \qquad \text{for } n = 1, \ldots, N \tag{15g}$$

The state transition matrices in Eq. (15b) can be expressed, for $\boldsymbol{r} \in R^3$, with the following expressions (with $\boldsymbol{I}$ being the identity matrix and $\boldsymbol{0}$ the zero matrix):

$$\boldsymbol{\Phi}_{\boldsymbol{r}}^{[n]} = \begin{bmatrix} \boldsymbol{I}_3 & \Delta t^{[n]} \boldsymbol{I}_3 \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{I}_3 \end{bmatrix} \quad \boldsymbol{\Psi}_{\boldsymbol{r}}^{[n]} = \begin{bmatrix} \left(\Delta t^{[n]}\right)^2 \boldsymbol{I}_3 \\ \Delta t^{[n]} \boldsymbol{I}_3 \end{bmatrix} \tag{16}$$

It is worth noting at this point that if the keep-out zone associated with the RSO is a single convex shape then, this sequential convex optimization approach enjoys guaranteed convergence.[10,20]

Solution's Initialization

To obtain a convex approximation of the problem a reference trajectory $\tilde{\boldsymbol{r}}$ is required. This trajectory is constructed using the previous iteration's solution ${}^{[k-1]}\boldsymbol{F}^{\star} \rightarrow {}^{[k]}\tilde{\boldsymbol{r}}$. When the problem is solved for the first time, a previous solution is not available. To avoid having to devise an initial guess, the constraint in Eq. (15d) can be dropped when the problem is solved for the first time. This first solution can then be used to produce the initial reference trajectory $\tilde{\boldsymbol{r}}$ and linearize the keep-out zone constraints. Additionally, if this initial solution satisfies the constraints, then the global optimal solution has already been found (no need to iterate).

Infeasibility Resolution

The convexification of the constraints, as in Eq. (15d), can make the problem infeasible. In that case, the constraint in Eq. (15d) can be removed and converted into a penalty, modifying the cost function in Eq. (15a) as follows:

$$J = \sum_{n=1}^{N} {\boldsymbol{F}^{[n]}}^T \boldsymbol{W}_{\boldsymbol{F}} \boldsymbol{F}^{[n]} \Delta t^{[n]} + \sum_{n=1}^{N} \sum_{i=1}^{P} \left\| -\tilde{d}_i^{[n]} - \hat{\boldsymbol{d}}_i^{[n]} \left(\boldsymbol{r}^{[n]} - \tilde{\boldsymbol{r}}^{[n]} \right) \right\|_1^+ \tag{17a}$$

$$\|x\|_1^+ = \begin{cases} x, \text{ for } x \geq 0 \\ 0, \text{ for } x < 0 \end{cases} \tag{17b}$$

When the linearized problem re-enters the feasible region, the penalty can be dropped and the constraint reinstated.

STEP 2: INTERNAL RE-CONFIGURATION OPTIMIZATION

The first step of the optimization, the system-wide translation optimization, generates locally optimal forces to be applied to the system's center-of-mass $\boldsymbol{F}^\star$. With $\boldsymbol{F}^\star$, a locally optimal center-of-mass trajectory, $\boldsymbol{r}^\star$, is derived. Once the system-wide optimization is completed, it is time to solve the internal re-configuration (consisting of the manipulator re-configuration and chaser re-orientation). The challenges in this optimization step emanate from the nonlinear attitude kinematics and nonlinear manipulator dynamics. As in the system-wide translation optimization, a sequential convex optimization approach will be used to overcome these difficulties.

To isolate the internal re-configuration, we use a reference frame anchored at the chaser's center-of-mass. In this non-inertial frame, the multibody system behaves as a translation-floating system, where the system's linear momentum is constant. By using a non-inertial frame, a set of inertial forces, derived from the frame's acceleration, appear. These inertial forces correspond to the forces applied to the center-of-mass $\boldsymbol{F}^\star$ but mapped into their equivalent generalized joint forces (torques at the base-spacecraft and torques at the manipulator's joints).

The equations of motion of a spacecraft-manipulator system, with respect to an inertial frame, can be written as in Eq. (18), with $\boldsymbol{r}_0$ denoting the chaser's position, $\boldsymbol{q}_0$ the chaser's attitude quaternion, $\boldsymbol{\theta}$ the manipulator joint deflections, $\boldsymbol{u}$ the generalized velocities and $\boldsymbol{M}$ the generalized forces. The generalized velocities $\boldsymbol{u}$ are composed of the chaser velocities (linear $\dot{\boldsymbol{r}}_0$ and angular ω_0) and manipulator joint velocities $\boldsymbol{\omega}_m$. Similarly, the generalized forces are composed of the chaser forces (linear $\boldsymbol{F}_0$ and angular $\boldsymbol{\tau}_0$) and the manipulator joint torques $\boldsymbol{\tau}_m$. Finally, $\boldsymbol{H}$ represents the inertia matrix and $\boldsymbol{C}$ the convective inertia matrix.

$$\boldsymbol{H}\left(\boldsymbol{q}_0, \boldsymbol{\theta}\right)\dot{\boldsymbol{u}} + \boldsymbol{C}\left(\boldsymbol{q}_0, \boldsymbol{\theta}, \boldsymbol{u}\right)\boldsymbol{u} = \boldsymbol{M} \tag{18a}$$

$$\boldsymbol{u} = \begin{bmatrix}\dot{\boldsymbol{r}}_0 & \boldsymbol{\omega}_0 & \boldsymbol{\omega}_m\end{bmatrix}^T \tag{18b}$$

$$\boldsymbol{M} = \begin{bmatrix}\boldsymbol{F}_0 & \boldsymbol{\tau}_0 & \boldsymbol{\tau}_m\end{bmatrix}^T \tag{18c}$$

The thrusters of the base-spacecraft are the only actuators that are able to impart linear momentum into the system. It is thus clear that the center-of-mass forces $\boldsymbol{F}^\star$, found during the system-wide translation optimization, will be actuated by the base-spacecraft thrusters.

$$\boldsymbol{F}_0 = \boldsymbol{F}^\star \tag{19}$$

If the equations of motion presented in Eq. (18) are used, setting $\boldsymbol{F}_0 = \boldsymbol{F}^\star$, then, linear momentum is imparted into the system. This imparted momentum displaces the center-of-mass, moving it as described by Eq. (1), thus recovering the $\boldsymbol{r}^\star$ trajectory. During this already predicted motion, the system will still exhibit internal reconfiguration, even when the rest of the forces are set to $\boldsymbol{\tau}_0 = \boldsymbol{\tau}_m = 0$. The system's dynamic coupling causes the system to internally react to the base-spacecraft forces.

By using a non-inertial reference frame, anchored at the system's center-of-mass, the system-wide translation can be discounted, placing the sole focus on the internal re-configuration. A set of inertial forces $\boldsymbol{M}_\mathcal{I}$ appear when we use this non-inertial frame. These inertial forces $\boldsymbol{M}_\mathcal{I}$ reflect the internal reaction of the system for a given center-of-mass acceleration.

$$\boldsymbol{H}\dot{\boldsymbol{u}} + \boldsymbol{C}\boldsymbol{u} = \boldsymbol{M} + \boldsymbol{M}_\mathcal{I} \tag{20}$$

The inertial forces $\boldsymbol{M}_{\mathcal{I}}$ are the opposite of $\boldsymbol{F}^{\star}$ when mapped into joint-space forces. These joint-space forces can be computed using the center-of-mass Jacobian $\boldsymbol{J}_{\mathrm{CoM}}$ and by exploiting the kineto-static duality.[21] The Jacobians and masses of each of the L links are denoted by $\boldsymbol{J}_l$ and m_l respectively.

$$\boldsymbol{M}_{\mathcal{I}} = -\boldsymbol{J}_{\mathrm{CoM}}^{T}\boldsymbol{F}^{\star} \tag{21a}$$

$$\boldsymbol{J}_{\mathrm{CoM}}\left(\boldsymbol{q}_0, \boldsymbol{\theta}\right) = \frac{\sum_{l=0}^{L} \boldsymbol{J}_l\left(\boldsymbol{q}_0, \boldsymbol{\theta}\right) m_l}{\sum_{l=0}^{L} m_l} \tag{21b}$$

Although it is not readily apparent from Eq. (21), it follows that $\boldsymbol{F}_{0\mathcal{I}} = -\boldsymbol{F}^{\star}$ (the thrusters mounted on the chaser are the only actuators able to impart any linear momentum). Therefore it is clear that when $\boldsymbol{M} = -\boldsymbol{M}_{\mathcal{I}}$ the system moves in bulk along $\boldsymbol{r}^{\star}$, but maintains its internal configuration (compensating the inertial forces).

As the base-spacecraft forces have been already determined and are thus fixed, the equations of motion, expressed in the non-inertial frame, can be re-written using a reduced generalized velocity $\overline{\boldsymbol{u}}$ and force $\overline{\boldsymbol{M}}$.

$$\boldsymbol{H}\begin{bmatrix}\ddot{\boldsymbol{r}}_0 \\ \dot{\overline{\boldsymbol{u}}}\end{bmatrix} + \boldsymbol{C}\begin{bmatrix}\dot{\boldsymbol{r}}_0 \\ \overline{\boldsymbol{u}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{F}_0 - \boldsymbol{F}^{\star} \\ \overline{\boldsymbol{M}} + \overline{\boldsymbol{M}}_{\mathcal{I}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{0} \\ \overline{\boldsymbol{M}} - \overline{\boldsymbol{J}}_{\mathrm{CoM}}^{T}\boldsymbol{F}^{\star}\end{bmatrix} \tag{22a}$$

$$\overline{\boldsymbol{u}} = \begin{bmatrix}\boldsymbol{\omega}_0 & \boldsymbol{\omega}_m\end{bmatrix}^{T} \tag{22b}$$

$$\overline{\boldsymbol{M}} = \begin{bmatrix}\boldsymbol{\tau}_0 & \boldsymbol{\tau}_m\end{bmatrix}^{T} \tag{22c}$$

The chaser linear motion reaction $\ddot{\boldsymbol{r}}_0$ can be found by solving the top part of Eq. (22a). By integrating this reaction, the chaser linear velocity $\dot{\boldsymbol{r}}_0$ and position $\boldsymbol{r}_0$ with respect to the center-of-mass can be obtained.

The cost function to be minimized during this sub-maneuver is chosen as the quadratic control effort associated with $\overline{\boldsymbol{M}}$, with $\boldsymbol{W}_{\overline{\boldsymbol{M}}}$ denoting a positive definite weight matrix.

$$J = \int_{t_0}^{t_f} \overline{\boldsymbol{M}}^{T}\boldsymbol{W}_{\overline{\boldsymbol{M}}}\overline{\boldsymbol{M}}\,dt \tag{23}$$

During the reconfiguration maneuver, the joint deflections are constrained within an upper and lower bound, imposing the following constraint:

$$\boldsymbol{\theta}_{\min} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_{\max} \tag{24}$$

The manipulator's joint and base-spacecraft torques are also limited.

$$\boldsymbol{\tau}_{m\,\min} \leq \boldsymbol{\tau}_m \leq \boldsymbol{\tau}_{m\,\max} \tag{25}$$

$$\left\|\boldsymbol{\tau}_0\right\|_2 \leq \tau_{0\,\max} \tag{26}$$

The terminal configuration of the manipulator has been pre-set and thus can also be imposed as a terminal equality constraint.

$$\boldsymbol{\theta}\left(t_f\right) = \boldsymbol{\theta}_f \tag{27}$$

A line-of-sight constraint on the chaser may also be imposed. During the maneuver it may be desirable to keep the chaser oriented towards a certain direction (*e.g.*, to keep the target RSO within the field-of-view of its navigation sensors).

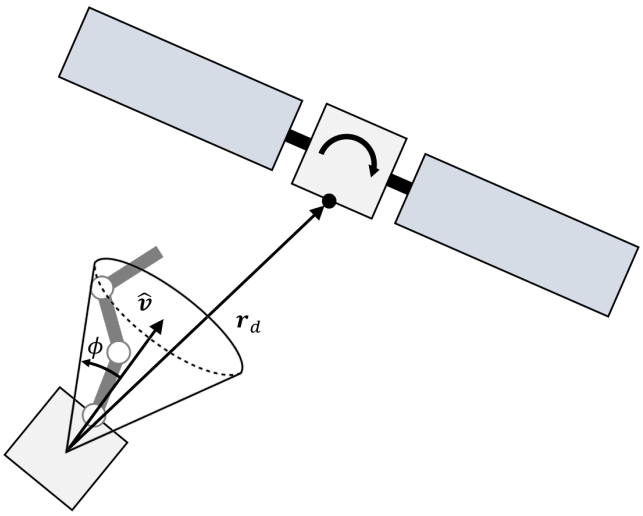


Figure 4. Notional Representation of the Line-of-Sight Constraint.

Convex formulations of line-of-sight constraint have been found using semidefinite matrices[22] or quaternion quadratic formulations.[23] A new explicitly convex quadratic formulation is presented here. The quadratic formulation reported by Lee[23] uses indefinite matrices, while our formulation use semidefinite matrices and results in an explicitly convex inequality constraint.

A notional representation of a line-of-sight constraint is shown in Figure 4 and can be mathematically expressed as follows:

$$\boldsymbol{r}_d \cdot \hat{\boldsymbol{v}} \geq \|\boldsymbol{r}_d\|_2 \cos\phi \tag{28a}$$

$$\boldsymbol{r}_d = \boldsymbol{r}_g - \boldsymbol{r} \tag{28b}$$

In Eq. (28a), $\hat{\boldsymbol{v}}$ denotes a vector with a constant orientation with respect to the chaser's body frame (*i.e.*, describing the boresight of a chaser's sensor) and $\boldsymbol{r}_d$ denotes the vector where $\hat{\boldsymbol{v}}$ would be ideally be pointing towards.

Usually, $\hat{\boldsymbol{v}}$ is known in a body fixed frame $\hat{\boldsymbol{v}}^{\mathcal{B}}$, while $\boldsymbol{r}_d$ is known on an inertial frame $\boldsymbol{r}_d^{\mathcal{I}}$. Using the attitude quaternion $\boldsymbol{q}_0$ one can rotate the $\boldsymbol{r}_d^{\mathcal{I}}$ vector and express it in the chaser's body frame $\boldsymbol{r}_d^{\mathcal{B}}$.

$$\boldsymbol{r}_d^{\mathcal{B}} = \boldsymbol{q}_0^* \otimes \hat{\boldsymbol{r}_d}^{\mathcal{I}} \otimes \boldsymbol{q}_0 \tag{29}$$

Using the properties of the bilinear quaternion multiplication, the line-of-sight constraint can be expressed by the following quadratic form (with $[\cdot]_L$ and $[\cdot]_R$ denoting the left and right-hand side matrix equivalents of the quaternion multiplication):

$$-\boldsymbol{q}_0^T \boldsymbol{A} \boldsymbol{q}_0 \geq \|\boldsymbol{r}_d\|_2 \cos\phi \tag{30a}$$

$$\boldsymbol{A} = \left[\hat{\boldsymbol{v}}^{\mathcal{B}}\right]_R \left[\boldsymbol{r}_d^{\mathcal{I}}\right]_L \tag{30b}$$

$$\boldsymbol{p} \otimes \boldsymbol{q} = [\boldsymbol{p}]_L \, \boldsymbol{q} = [\boldsymbol{q}]_R \, \boldsymbol{p} \tag{30c}$$

The $\boldsymbol{A}$ matrix in Eq. (30a) is indefinite and masks the convex nature of the constraint. Here we show how to express this line-of-sight constraint with an equivalent explicitly convex quadratic form.

The $\boldsymbol{A}$ matrix can be decomposed into a positive semidefinite matrix $\boldsymbol{A}^+$ and a negative semidefinite one $\boldsymbol{A}^-$. Each of these matrices can be constructed by using either the positive or negative eigenvalues of the $\boldsymbol{A}$ matrix.

If $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{V}$ is a square matrix containing its eigenvectors, then we can obtain $\boldsymbol{A}^+$ and $\boldsymbol{A}^-$ by collecting all the positive eigenvalues of $\boldsymbol{\Lambda}$ in $\boldsymbol{\Lambda}^+$, and all the negative ones in $\boldsymbol{\Lambda}^-$ as follows:

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T \tag{31a}$$

$$\boldsymbol{A} = \boldsymbol{V}\left(\boldsymbol{\Lambda}^+ + \boldsymbol{\Lambda}^-\right)\boldsymbol{V}^T \tag{31b}$$

$$\boldsymbol{A}^+ = \boldsymbol{V}\boldsymbol{\Lambda}^+\boldsymbol{V}^T \tag{31c}$$

$$\boldsymbol{A}^- = \boldsymbol{V}\boldsymbol{\Lambda}^-\boldsymbol{V}^T \tag{31d}$$

$$\boldsymbol{A} = \boldsymbol{A}^+ + \boldsymbol{A}^- \tag{31e}$$

The following identities can then be obtained:

$$\boldsymbol{q}_0^T\boldsymbol{A}^+\boldsymbol{q}_0 = \|\boldsymbol{r}_d\|_2\frac{1-\cos\theta}{2} \tag{32a}$$

$$\boldsymbol{q}_0^T\boldsymbol{A}^-\boldsymbol{q}_0 = \|\boldsymbol{r}_2\|_2\frac{-1-\cos\theta}{2} \tag{32b}$$

The line-of-sight constraint can be formulated using these two equivalent quadratic expressions.

$$-2\boldsymbol{q}_0^T\boldsymbol{A}^+\boldsymbol{q}_0 + \|\boldsymbol{r}_d\|_2\left(1-\cos\phi\right) \geq 0 \tag{33a}$$

$$-2\boldsymbol{q}_0^T\boldsymbol{A}^-\boldsymbol{q}_0 - \|\boldsymbol{r}_d\|_2\left(1+\cos\phi\right) \geq 0 \tag{33b}$$

Note that the expression in Eq. (33a) is concave and that it is subject to an inequality of the type $\geq$ and thus it can be expressed using a canonical convex inequality constraint as follows: *

$$2\boldsymbol{q}_0^T\boldsymbol{A}^+\boldsymbol{q}_0 + \|\boldsymbol{r}_d\|_2\left(\cos\phi-1\right) \leq 0 \tag{34}$$

It is readily apparent that Eq. (34) is convex. This expression can help formulate convex optimization problems in solvers that require explicitly convex expressions (such as CVX[24]).

The final orientation of the chaser needs to be aligned with a certain axis $\hat{\boldsymbol{v}}_f$, that is dependent on the final position of the center-of-mass with respect to the grappling fixture. The chaser is free to rotate around this axis. This constraint is analogous to the line-of-sight constraint but with $\phi = 0$.

$$2\boldsymbol{q}_0^T\boldsymbol{A}^+\boldsymbol{q}_0 \leq 0 \tag{35}$$

*As an aside, Eq. (33b) can be used to express an attitude exclusion zone of the form

$$\boldsymbol{r}\cdot\boldsymbol{v} \leq \|\boldsymbol{v}\|\|\boldsymbol{r}\|\cos\phi$$

with the following canonical convex inequality constraint:

$$-2\boldsymbol{q}^T\boldsymbol{A}^-\boldsymbol{q} - \|\boldsymbol{v}\|\|\boldsymbol{r}\|\left(1+\cos\phi\right) \leq 0$$

To complete the optimization problem formulation, we need to include the quaternion differential kinematics equation.

$$\dot{\boldsymbol{q}}_0 = \frac{1}{2}\boldsymbol{q}_0 \otimes \boldsymbol{\omega}_0 \tag{36}$$

The entire optimization problem can be formulated as follows:

$$\text{minimize: } \int_{t_0}^{t_f} \overline{\boldsymbol{M}}^T \boldsymbol{W}_{\overline{\boldsymbol{M}}} \overline{\boldsymbol{M}} dt \tag{37a}$$

$$\text{subject to: } \boldsymbol{H}\begin{bmatrix}\ddot{\boldsymbol{r}}_0 \\ \dot{\overline{\boldsymbol{u}}}\end{bmatrix} + \boldsymbol{C}\begin{bmatrix}\dot{\boldsymbol{r}}_0 \\ \overline{\boldsymbol{u}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{0} \\ \overline{\boldsymbol{M}} - \overline{\boldsymbol{J}}_{\mathrm{CoM}}^T \boldsymbol{F}\end{bmatrix} \tag{37b}$$

$$\dot{\boldsymbol{q}}_0 = \frac{1}{2}\boldsymbol{q}_0 \otimes \boldsymbol{\omega}_0 \tag{37c}$$

$$\boldsymbol{\theta}_{\min} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_{\max} \tag{37d}$$

$$\boldsymbol{\tau}_{m\,\min} \leq \boldsymbol{\tau}_m \leq \boldsymbol{\tau}_{m\,\max} \tag{37e}$$

$$\|\boldsymbol{\tau}_0\|_2 \leq \boldsymbol{\tau}_{0\,\max} \tag{37f}$$

$$2\boldsymbol{q}_0^T \boldsymbol{A}^+ \boldsymbol{q}_0 + \|\boldsymbol{r}_d\|_2 (\cos\phi - 1) \leq 0 \tag{37g}$$

$$2\boldsymbol{q}_0^T \boldsymbol{A}_f^+ \boldsymbol{q}_0 \leq 0 \tag{37h}$$

$$\boldsymbol{\theta}(t_f) = \boldsymbol{\theta}_f \tag{37i}$$

To problem is discretized by direct transcription with N nodes. In this case, the optimization variable will be the generalized accelerations $\dot{\overline{\boldsymbol{u}}}$. Therefore, the discretized accelerations will be piecewise constant $\dot{\overline{\boldsymbol{u}}}^{[n]}$, the velocities piecewise linear $\overline{\boldsymbol{u}}^{[n]}$, and the generalized forces piecewise nonlinear $\overline{\boldsymbol{M}}^{[n]}$.

A simple trapezoidal integration scheme can be used to obtain the cost.

$$J = \sum_{n=0}^{N-1} \frac{\overline{\boldsymbol{M}}^{[n+1]^T} \boldsymbol{W}_{\overline{\boldsymbol{M}}} \overline{\boldsymbol{M}}^{[n+1]} - \overline{\boldsymbol{M}}^{[n]^T} \boldsymbol{W}_{\overline{\boldsymbol{M}}} \overline{\boldsymbol{M}}^{[n]}}{2} \Delta t^{[n]} \tag{38}$$

The velocities $\overline{\boldsymbol{u}}$ can be propagated using state transition matrices.

$$\overline{\boldsymbol{u}}^{[n+1]} = \boldsymbol{\Phi}_{\boldsymbol{u}} \overline{\boldsymbol{u}}^{[n]} + \boldsymbol{\Theta}_{\boldsymbol{u}}^{[n]} \dot{\overline{\boldsymbol{u}}}^{[n]} \tag{39a}$$

$$\overline{\boldsymbol{u}}^{[n+1]} = \boldsymbol{\Phi}_{\boldsymbol{u}}^n \overline{\boldsymbol{u}}^{[1]} + \sum_{i=1}^{n} \boldsymbol{\Phi}_{\boldsymbol{u}}^{i-1} \boldsymbol{\Theta}_{\boldsymbol{u}}^{[i]} \overline{\boldsymbol{u}}^{[i]} \tag{39b}$$

$$\boldsymbol{\Phi}_{\boldsymbol{u}} = \boldsymbol{I} \tag{39c}$$

$$\boldsymbol{\Theta}_{\boldsymbol{u}}^{[n]} = \boldsymbol{I} \Delta t^{[n]} \tag{39d}$$

An analogous approach can be used to propagate the displacements of manipulator's joints.

$$\boldsymbol{\theta}^{[n+1]} = \boldsymbol{\Phi}_{\boldsymbol{\theta}} \boldsymbol{\theta}^{[n]} + \boldsymbol{\Theta}_{\boldsymbol{\theta}}^{[n]} \begin{bmatrix}\boldsymbol{\omega}_m^{[n]} \\ \dot{\boldsymbol{\omega}}_m^{[n]}\end{bmatrix} \tag{40a}$$

$$\boldsymbol{\theta}^{[n+1]} = \boldsymbol{\Phi}_{\boldsymbol{\theta}}^n \boldsymbol{\theta}^{[1]} + \sum_{i=1}^{n} \boldsymbol{\Phi}_{\boldsymbol{\theta}}^{i-1} \boldsymbol{\Theta}_{\boldsymbol{\theta}}^{[i]} \begin{bmatrix}\boldsymbol{\omega}_m^{[n]} \\ \dot{\boldsymbol{\omega}}_m^{[n]}\end{bmatrix} \tag{40b}$$

$$\boldsymbol{\Phi}_{\boldsymbol{\theta}} = \boldsymbol{I} \tag{40c}$$

$$\boldsymbol{\Theta}_{\boldsymbol{\theta}}^{[n]} = \begin{bmatrix} \boldsymbol{I}\Delta t^{[n]} & \boldsymbol{I}\left(\Delta t^{[n]}\right)^2 \end{bmatrix} \tag{40d}$$

The propagation of the nonlinear quaternion kinematics is more problematic. As the angular velocity is piecewise linear, the forward integration of the quaternion differential kinematics equation can be approximated using the following expression:

$$\boldsymbol{\omega}_0^{[n+1/2]} = \frac{\boldsymbol{\omega}_0^{[n+1]} + \boldsymbol{\omega}_0^{[n]}}{2} \tag{41a}$$

$$\boldsymbol{q}_0^{[n+1]} \approx \boldsymbol{q}_0^{[n]} \otimes \begin{bmatrix} \frac{\boldsymbol{\omega}_0^{[n+1/2]}}{\left\|\boldsymbol{\omega}_0^{[n+1/2]}\right\|_2} \sin\left\|\boldsymbol{\omega}_0^{[n+1/2]}\right\|_2 \frac{\Delta t^{[n]}}{2} \\ \cos\left\|\boldsymbol{\omega}_0^{[n+1/2]}\right\|_2 \frac{\Delta t^{[n]}}{2} \end{bmatrix} = \boldsymbol{q}_0^{[n]} \otimes \boldsymbol{q}\left\{\boldsymbol{\omega}_0^{[n+1/2]}\Delta t^{[n]}\right\} \tag{41b}$$

$$\boldsymbol{q}_0^{[n+1]} \approx \boldsymbol{q}_0^{[1]} \otimes \boldsymbol{q}\left\{\boldsymbol{\omega}_0^{[1+1/2]}\Delta t^{[n]}\right\} \otimes \ldots \otimes \boldsymbol{q}\left\{\boldsymbol{\omega}_0^{[n+1/2]}\Delta t^{[n]}\right\} = f_{\Delta \boldsymbol{q}}^{[n]}\left(\boldsymbol{q}_0, \boldsymbol{\omega}_0^{[n]}, \ldots, \boldsymbol{\omega}_0^{[n+1]}\right) \tag{41c}$$

Despite the approximate nature of Eq. (41c), the quaternion propagation remains nonlinear. To overcome this nonlinearity, a sequential convex optimization approach is used. The nonlinear kinematics are linearized, allowing the approximate convex formulation of the problem. The problem is repeatedly solved, using the last iteration's solution to repeatedly linearize the nonlinear kinematics.

Using a Taylor expansion around a particular trajectory $\tilde{\boldsymbol{\omega}}_0$, truncating to keep only the linear terms, we obtain a linear approximation of the function $f_{\Delta \boldsymbol{q}}$ introduced in Eq. (41c).

$$f_{\Delta \boldsymbol{q}} = f_{\Delta \boldsymbol{q}}\left(\tilde{\boldsymbol{\omega}}_0\right) + \nabla f_{\Delta \boldsymbol{q}}\left(\tilde{\boldsymbol{\omega}}_0\right)\left(\boldsymbol{\omega}_0 - \tilde{\boldsymbol{\omega}}_0\right) + \frac{1}{2!}\left(\boldsymbol{\omega}_0 - \tilde{\boldsymbol{\omega}}_0\right)^T \nabla^2 f_{\Delta \boldsymbol{q}}\left(\tilde{\boldsymbol{\omega}}_0\right)\left(\boldsymbol{\omega}_0 - \tilde{\boldsymbol{\omega}}_0\right) + \ldots \tag{42a}$$

$$f_{\Delta \boldsymbol{q}} \approx f_{\Delta \boldsymbol{q}}\left(\tilde{\boldsymbol{\omega}}_0\right) + \nabla f_{\Delta \boldsymbol{q}}\left(\tilde{\boldsymbol{\omega}}_0\right)\left(\boldsymbol{\omega}_0 - \tilde{\boldsymbol{\omega}}_0\right) \tag{42b}$$

Later on, a second order correction will be introduced to improve the robustness and convergence properties of the optimization procedure.

An analogous procedure is used to approximate the nonlinear system dynamics. First, the dynamics are approximated using a Taylor expansion and then the expansion is truncated to keep only the linear terms.

$$\boldsymbol{H}\begin{bmatrix} \ddot{\boldsymbol{r}}_0 \\ \dot{\overline{\boldsymbol{u}}} \end{bmatrix} + \boldsymbol{C}\begin{bmatrix} \dot{\boldsymbol{r}}_0 \\ \overline{\boldsymbol{u}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{J}}_{\mathrm{CoM}}^T \boldsymbol{F} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{M}} \end{bmatrix} \tag{43a}$$

$$\overline{\boldsymbol{M}} = f_{\boldsymbol{M}_c}\left(\boldsymbol{q}_0, \boldsymbol{\theta}, \overline{\boldsymbol{u}}\right) = f_{\boldsymbol{M}_c}\left(\dot{\overline{\boldsymbol{u}}}\right) \tag{43b}$$

$$f_{\boldsymbol{M}_c} = f_{\boldsymbol{M}_c}\left(\tilde{\dot{\overline{\boldsymbol{u}}}}\right) + \nabla f_{\boldsymbol{M}_c}\left(\tilde{\dot{\overline{\boldsymbol{u}}}}\right)\left(\dot{\overline{\boldsymbol{u}}} - \tilde{\dot{\overline{\boldsymbol{u}}}}\right) + \frac{1}{2!}\left(\dot{\overline{\boldsymbol{u}}} - \tilde{\dot{\overline{\boldsymbol{u}}}}\right)^T \nabla^2 f_{\boldsymbol{M}_c}\left(\tilde{\dot{\overline{\boldsymbol{u}}}}\right)\left(\dot{\overline{\boldsymbol{u}}} - \tilde{\dot{\overline{\boldsymbol{u}}}}\right) + \ldots \tag{43c}$$

$$\overline{\boldsymbol{M}} \approx f_{\boldsymbol{M}_c}\left(\tilde{\dot{\overline{\boldsymbol{u}}}}\right) + \nabla f_{\boldsymbol{M}_c}\left(\tilde{\dot{\overline{\boldsymbol{u}}}}\right)\left(\dot{\overline{\boldsymbol{u}}} - \tilde{\dot{\overline{\boldsymbol{u}}}}\right) \tag{43d}$$

The original problem has now been discretized and linearized. By sequentially solving this convex approximation of the original problem we can obtain a local solution to the original non-convex problem. An important aspect of a sequential convex optimization is to impose trust regions. The trust regions ensure that the solution will remain in a region where the approximations are still valid. In this case trust regions around $\boldsymbol{\theta}$, $\overline{\boldsymbol{u}}$ and $\dot{\overline{\boldsymbol{u}}}$ are imposed.

$$\left\|\boldsymbol{\theta}^{[n]} - \tilde{\boldsymbol{\theta}}^{[n]}\right\|_1 \leq \rho_{\boldsymbol{\theta}} \tag{44a}$$

$$\left\| \overline{\boldsymbol{u}}^{[n]} - \tilde{\overline{\boldsymbol{u}}}^{[n]} \right\|_1 \leq \rho_{\overline{\boldsymbol{u}}} \tag{44b}$$

$$\left\| \dot{\overline{\boldsymbol{u}}}^{[n]} - \tilde{\dot{\overline{\boldsymbol{u}}}}^{[n]} \right\|_1 \leq \rho_{\dot{\overline{\boldsymbol{u}}}} \tag{44c}$$

Finally, the discretized convex approximation of the problem can be expressed with the following expressions:

$$\text{minimize: } \sum_{n=0}^{N-1} \frac{\overline{\boldsymbol{M}}^{[n+1]^T} \boldsymbol{W}_{\overline{\boldsymbol{M}}} \overline{\boldsymbol{M}}^{[n+1]} - \overline{\boldsymbol{M}}^{[n]^T} \boldsymbol{W}_{\overline{\boldsymbol{M}}} \overline{\boldsymbol{M}}^{[n]}}{2} \Delta t^{[n]} \tag{45a}$$

$$\text{subject to: } \overline{\boldsymbol{M}}^{[n]} = f_{\boldsymbol{M}_c}^{[n]} \left(\tilde{\dot{\overline{\boldsymbol{u}}}} \right) + \nabla f_{\boldsymbol{M}_c}^{[n]} \left(\tilde{\dot{\overline{\boldsymbol{u}}}} \right) \left(\dot{\overline{\boldsymbol{u}}} - \tilde{\dot{\overline{\boldsymbol{u}}}} \right) \qquad \text{for } n = 1, \ldots, N \tag{45b}$$

$$\overline{\boldsymbol{u}}^{[n+1]} = \boldsymbol{\Phi}^n \overline{\boldsymbol{u}}^{[1]} + \sum_{i=1}^{n} \boldsymbol{\Phi}_{\boldsymbol{u}}^{i-1} \boldsymbol{\Theta}_{\boldsymbol{u}}^{[i]} \overline{\boldsymbol{u}}^{[i]} \qquad \text{for } n = 1, \ldots, N-1 \tag{45c}$$

$$\boldsymbol{\theta}^{[n+1]} = \boldsymbol{\Phi}_{\boldsymbol{\theta}}^n \boldsymbol{\theta}^{[1]} + \sum_{i=1}^{n} \boldsymbol{\Phi}_{\boldsymbol{\theta}}^{i-1} \boldsymbol{\Theta}_{\boldsymbol{\theta}}^{[i]} \begin{bmatrix} \boldsymbol{\omega}_m^{[n]} \\ \dot{\boldsymbol{\omega}}_m^{[n]} \end{bmatrix} \qquad \text{for } n = 1, \ldots, N-1 \tag{45d}$$

$$\boldsymbol{q}_0^{[n+1]} = f_{\Delta \boldsymbol{q}}^{[n]} \left(\tilde{\boldsymbol{\omega}}_0 \right) + \nabla f_{\Delta \boldsymbol{q}}^{[n]} \left(\tilde{\boldsymbol{\omega}}_0 \right) \left(\boldsymbol{\omega}_0 - \tilde{\boldsymbol{\omega}}_0 \right) \qquad \text{for } n = 1, \ldots, N-1 \tag{45e}$$

$$\boldsymbol{\theta}_{\min} \leq \boldsymbol{\theta}^{[n]} \leq \boldsymbol{\theta}_{\max} \qquad \text{for } n = 1, \ldots, N \tag{45f}$$

$$\boldsymbol{\tau}_{m\,\min} \leq \boldsymbol{\tau}_m^{[n]} \leq \boldsymbol{\tau}_{m\,\max} \qquad \text{for } n = 1, \ldots, N \tag{45g}$$

$$\left\| \boldsymbol{\tau}_0^{[n]} \right\|_2 \leq \boldsymbol{\tau}_{0\,\max} \qquad \text{for } n = 1, \ldots, N \tag{45h}$$

$$2 \boldsymbol{q}_0^{[n]^T} \boldsymbol{A}^{+[n]} \boldsymbol{q}_0^{[n]} + \| \boldsymbol{r}_d^{[n]} \|_2 \left(\cos \phi - 1 \right) \leq 0 \qquad \text{for } n = 1, \ldots, N \tag{45i}$$

$$2 \boldsymbol{q}_0^{[n]^T} \boldsymbol{A}_f^{+} \boldsymbol{q}_0^{[n]} \leq 0 \qquad \text{for } n = N \tag{45j}$$

$$\boldsymbol{\theta}^{[n]} = \boldsymbol{\theta}_f \qquad \text{for } n = N \tag{45k}$$

$$\left\| \boldsymbol{\theta}^{[n]} - \tilde{\boldsymbol{\theta}}^{[n]} \right\|_1 \leq \rho_\theta \qquad \text{for } n = 1, \ldots, N \tag{45l}$$

$$\left\| \overline{\boldsymbol{u}}^{[n]} - \tilde{\overline{\boldsymbol{u}}}^{[n]} \right\|_1 \leq \rho_{\overline{\boldsymbol{u}}} \qquad \text{for } n = 1, \ldots, N \tag{45m}$$

$$\left\| \dot{\overline{\boldsymbol{u}}}^{[n]} - \tilde{\dot{\overline{\boldsymbol{u}}}}^{[n]} \right\|_1 \leq \rho_{\dot{\overline{\boldsymbol{u}}}} \qquad \text{for } n = 1, \ldots, N \tag{45n}$$

To improve the robustness and the convergence properties of the algorithm, an additional optimization step can be added. Once the approximated convex problem is solved, and the current iteration's optimal solution found ${}^{[k-1]}\dot{\boldsymbol{u}}^\star$, the problem can be resolved with the addition of a second-order correction to the linearized kinematics and dynamics.[10]

The second-order correction for the generalized forces can be computed as follows:

$$\overline{\boldsymbol{M}}^{[n]} = f_{\boldsymbol{M}_c}^{[n]} \left(\tilde{\dot{\overline{\boldsymbol{u}}}} \right) + \nabla f_{\boldsymbol{M}_c}^{[n]} \left(\tilde{\dot{\overline{\boldsymbol{u}}}} \right) \left(\dot{\overline{\boldsymbol{u}}} - \tilde{\dot{\overline{\boldsymbol{u}}}} \right) + f_{\boldsymbol{M}_c}^{\mathrm{SO}[n]} \left(\tilde{\dot{\boldsymbol{u}}}^n \right) \tag{46a}$$

$$f_{\boldsymbol{M}_c}^{\mathrm{SO}} \left(\tilde{\dot{\overline{\boldsymbol{u}}}} \right) = f_{\boldsymbol{M}_c} \left(\dot{\overline{\boldsymbol{u}}}^\star \right) - f_{\boldsymbol{M}_c} \left(\tilde{\dot{\overline{\boldsymbol{u}}}} \right) - \nabla f_{\boldsymbol{M}_c} \left(\tilde{\dot{\overline{\boldsymbol{u}}}} \right) \left(\dot{\overline{\boldsymbol{u}}}^\star - \tilde{\dot{\overline{\boldsymbol{u}}}} \right) \tag{46b}$$

The same can be applied to the quaternion differential kinematics linear approximation.

$$\boldsymbol{q}_0^{[n+1]} = f_{\Delta \boldsymbol{q}}^{[n]} \left(\tilde{\boldsymbol{\omega}}_0 \right) + \nabla f_{\Delta \boldsymbol{q}}^{[n]} \left(\tilde{\boldsymbol{\omega}}_0 \right) \left(\boldsymbol{\omega}_0 - \tilde{\boldsymbol{\omega}}_0 \right) + f_{\Delta \boldsymbol{q}}^{\mathrm{SO}[n]} \left(\tilde{\boldsymbol{\omega}}_0 \right) \tag{47a}$$

$$f^{\mathrm{SO}}_{\Delta \boldsymbol{q}}\left(\tilde{\boldsymbol{\omega}}_0\right) = f_{\Delta \boldsymbol{q}}\left(\boldsymbol{\omega}^{\star}_0\right) - f_{\Delta \boldsymbol{q}}\left(\tilde{\boldsymbol{\omega}}_0\right) - \nabla f_{\Delta \boldsymbol{q}}\left(\tilde{\boldsymbol{\omega}}_0\right)\left(\boldsymbol{\omega}^{\star}_0 - \tilde{\boldsymbol{\omega}}_0\right) \tag{47b}$$

The discretized problem with the second order correction can be formulated as follows:

$$\text{minimize: } \sum_{n=0}^{N-1} \frac{\overline{\boldsymbol{M}}^{[n+1]^T} \boldsymbol{W}_{\overline{\boldsymbol{M}}} \overline{\boldsymbol{M}}^{[n+1]} - \overline{\boldsymbol{M}}^{[n]^T} \boldsymbol{W}_{\overline{\boldsymbol{M}}} \overline{\boldsymbol{M}}^{[n]}}{2} \Delta t^{[n]} \tag{48a}$$

$$\text{subject to: } \overline{\boldsymbol{M}}^{[n]} = f^{[n]}_{\boldsymbol{M}_c}\left(\tilde{\dot{\overline{\boldsymbol{u}}}}\right) + \nabla f^{[n]}_{\boldsymbol{M}_c}\left(\tilde{\dot{\overline{\boldsymbol{u}}}}\right)\left(\dot{\overline{\boldsymbol{u}}} - \tilde{\dot{\overline{\boldsymbol{u}}}}\right) + f^{\mathrm{SO}[n]}_{\boldsymbol{M}_c} \qquad \text{for } n = 1, \ldots, N \tag{48b}$$

$$\overline{\boldsymbol{u}}^{[n+1]} = \boldsymbol{\Phi}^{n} \overline{\boldsymbol{u}}^{[1]} + \sum_{i=1}^{n} \boldsymbol{\Phi}^{i-1}_{\boldsymbol{u}} \boldsymbol{\Theta}^{[i]}_{\boldsymbol{u}} \overline{\boldsymbol{u}}^{[i]} \qquad \text{for } n = 1, \ldots, N-1 \tag{48c}$$

$$\boldsymbol{\theta}^{[n+1]} = \boldsymbol{\Phi}^{n}_{\boldsymbol{\theta}} \boldsymbol{\theta}^{[1]} + \sum_{i=1}^{n} \boldsymbol{\Phi}^{i-1}_{\boldsymbol{\theta}} \boldsymbol{\Theta}^{[i]}_{\boldsymbol{\theta}} \begin{bmatrix} \boldsymbol{\omega}^{[n]}_m \\ \dot{\boldsymbol{\omega}}^{[n]}_m \end{bmatrix} \qquad \text{for } n = 1, \ldots, N-1 \tag{48d}$$

$$\boldsymbol{q}^{[n+1]}_0 = f^{[n]}_{\Delta \boldsymbol{q}}\left(\tilde{\boldsymbol{\omega}}_0\right) + \nabla f^{[n]}_{\Delta \boldsymbol{q}}\left(\tilde{\boldsymbol{\omega}}_0\right)\left(\boldsymbol{\omega}_0 - \tilde{\boldsymbol{\omega}}_0\right) + f^{\mathrm{SO}[n]}_{\Delta \boldsymbol{q}} \qquad \text{for } n = 1, \ldots, N-1 \tag{48e}$$

and Eq. (45f)-(45n)

With this second-order correction step, each sequential convex optimization iteration consists of solving two convex optimization problems. The first one uses only the linear approximations (see Eq. (45)) and the second builds upon the solution of the first, and includes the second-order corrections (see Eq. (48)).

Initialization

The convex approximation of the problem relies on a linearization around a set trajectory $\tilde{\dot{\overline{\boldsymbol{u}}}}$. The first time the problem is solved there is no previous solution to rely on and thus an initial guess is required. Unlike the initialization of the system-wide translation sequential convex optimization, we cannot drop the dynamics, thus we need a feasible initial guess to start the optimization procedure. Generating a feasible initial guess can be a challenge. We use a simple manipulator motion (*i.e.*, minimum deflection from initial pose to final pose) and a chaser attitude that is perfectly aligned with $\boldsymbol{r}_d$ in order to generate this initial solution.

OVERVIEW OF THE ENTIRE TWO-STEP OPTIMIZATION PROCEDURE

A diagram of the overall optimization procedure is shown in Figure 5. The two optimization problems are solved sequentially, first the system-wide translation optimization is solved, resulting in the optimal forces $\boldsymbol{F}^{\star}$ applied at the system's center-of-mass. Using the results of the system-wide translation the second optimization step, tackling the internal re-configuration, is performed, resulting in the optimal base and manipulator's joints torques $\overline{\boldsymbol{M}}^{\star}$.

SIMULATION RESULTS

To validate the proposed approach a simulation example is provided. In this example, a chaser spacecraft equipped with a three degree-of-freedom robotic manipulator is tasked with the capture of a tumbling RSO.

A notional view of the target RSO and of the chaser is shown in Figure 6. The simulation parameters used for the simulation are provided in Tables 1-3. To solve the convex optimization problems we use CVX[24,25] with the SDPT3 solver.[26]

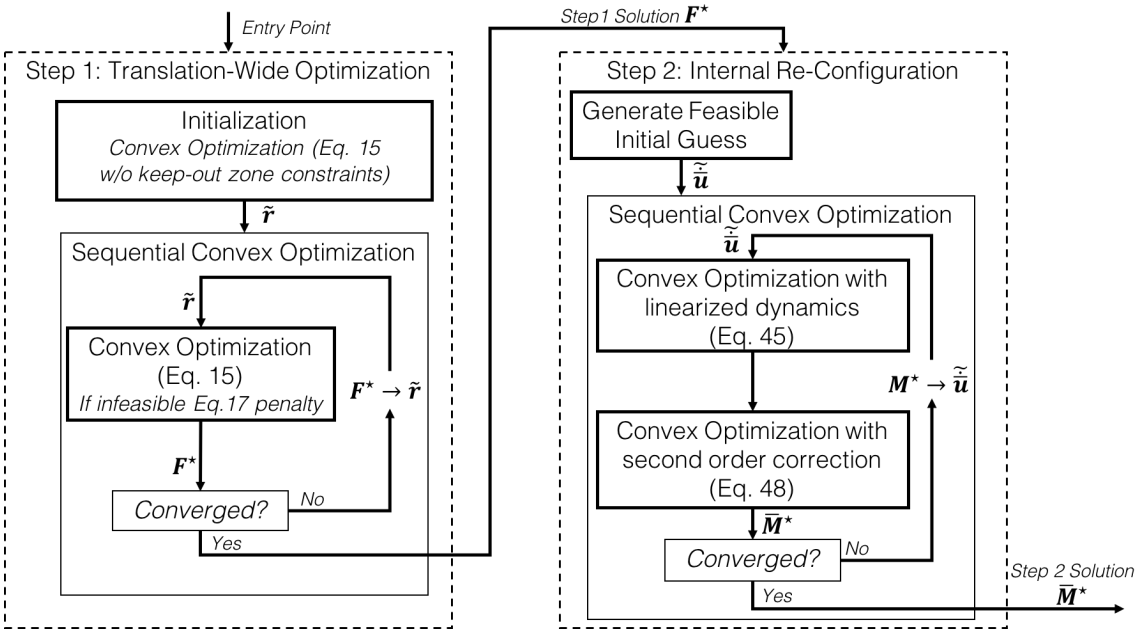


Figure 5. Notional Diagram of the Overall Optimization Procedure.

Table 1. RSO Numerical Simulation Parameters.

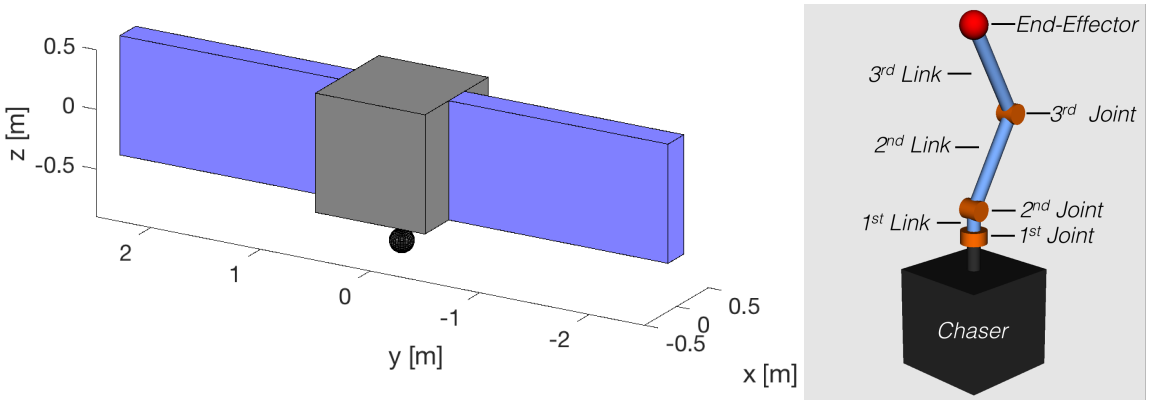
RSO Parameter	Value
RSO position	$[0 \quad 0 \quad 0]$ m
Grappling fixture	0.8 m along $-\boldsymbol{z}$
Initial angular velocity	$[1 \quad 1 \quad 1]^\circ\,\mathrm{s}^{-1}$
Mass	240 kg
Inertia	$J_{xx} = 140$, $J_{yy} = 36.9$, and $J_{zz} = 36.9\,\mathrm{kg\,m}^2$

Step 1: System-Wide Translation Results

The initial and final solutions for the system-wide translation optimization are provided in Figure 7. In this figure, the sphere R_{ko} used to enforce the keep-out zone constraints and the smaller sphere (in red), representing the final pose reach R_f, can be seen. As expected, the initial solution, which ignores the keep-out zone constraints, collides with the RSO (the initial conditions of the chaser were intentionally chosen to illustrate this particular effect). As the optimization progresses, and the keep-out zone constraints are taken into account, the trajectory bends to avoid colliding with the RSO. The converged collision-free solution is also shown in Figure 7.

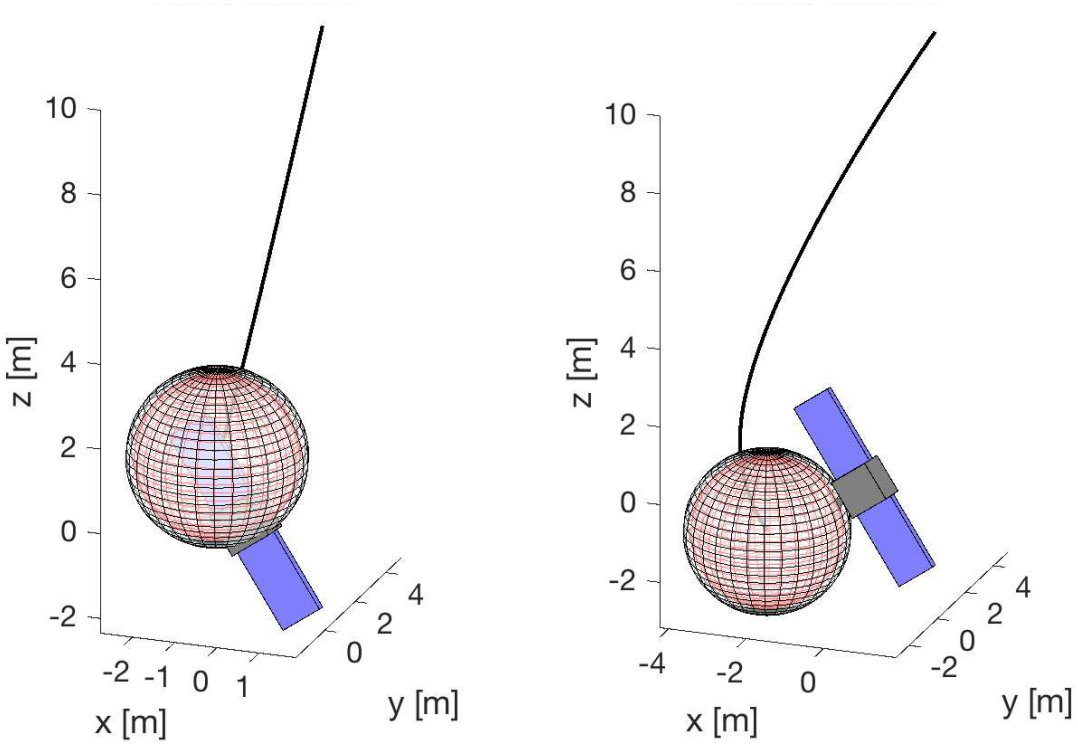
Step 2: Internal Re-Configuration Results

After completing the first optimization step, the internal re-configuration maneuver optimization is solved. Before starting to solve the problem an initial guess must be generated in order to provide the reference trajectory used to linearize the problem. In this case, the initial guess assumes a manipulator that deploys from its initial configuration to its final using the minimum joint deflection path. The chaser's attitude initial guess is chosen to continuously orient the chaser towards the grappling fixture (*i.e.*, perfect alignment during all the trajectory).



(a) RSO Model (b) Chaser Model

Figure 6. RSO and Chaser Model used During the Numerical Simulation.



(a) Initial Solution (b) Final Solution

Figure 7. Solutions of the System-Wide Translation.

Table 2. Chaser Numerical Simulation Parameters.

Chaser Parameter	Value
Chaser initial position	$[0 \quad 5 \quad 10]$ m
Chaser initial velocity	$[0 \quad 0 \quad 0]$ m s^{-1}
Chaser initial orientation	Pointing towards the RSO
Chaser initial angular velocity	$[0 \quad 0 \quad 0]^\circ$ s^{-1}
Mass of the base	105 kg
Inertia of the base	$J_{xx,yy,zz} = 9.3$ kg m^2
Mass of the links	$m_1 = 5$, $m_2 = 10$, and $m_3 = 10$ kg
Length of the links	$l_1 = 0.2$, $l_2 = 0.75$, and $l_3 = 0.75$ m
Initial pose $\boldsymbol{\theta}(t_0)$	$\theta_1 = \pi$, $\theta_2 = \frac{\pi}{2}$, and $\theta_3 = \frac{\pi}{2}$
Grasping pose $\boldsymbol{\theta}(t_f)$	$\theta_1 = -\pi$, $\theta_2 = \frac{\pi}{6}$, and $\theta_3 = \frac{\pi}{3}$
Grasping pose reach	$R_f = 1.85$ m
Keep-out sphere radius	$R_{\text{ko}} = 2.04$ m
Max. force	$F_{0\,\max} = 6.25$ N
Max. base torque	$\tau_{0\,\max} = 1$ N m
Max. joint torque	$\tau_{m\,\max} = 5$ N m
Max. joint deflections	$\theta_1 = \pm\pi$, $\theta_2 = \pm\frac{\pi}{2}$, and $\theta_3 = \pm 1.75$
Line-of-sight cone half-angle	$\phi = 15^\circ$

Table 3. Optimization Parameters.

Optimization Parameter	Value
Final time	$t_f = 60$ s
Number of nodes	$N = 101$
Step 1 trust region	$\rho_{\boldsymbol{r}} = 0.1$ m
Step 2 trust regions	$\rho_{\boldsymbol{\theta}} = 2^\circ$, $\rho_{\bar{\boldsymbol{u}}} = 2^\circ$ s^{-1}, and $\rho_{\dot{\bar{\boldsymbol{u}}}} = 0.2^\circ$ s^{-2}

To linearize the system dynamics it is assumed that the inertia and convective inertia matrices are constant (constructed using the last iteration's solution). The resulting linearized dynamics are shown in Eq. (49). The dynamic and kinematic properties of the spacecraft-manipulator system have been obtained by using the SPART toolkit.[27,28]

$$\tilde{\boldsymbol{H}} \begin{bmatrix} \ddot{\boldsymbol{r}}_0 \\ \dot{\bar{\boldsymbol{u}}} \end{bmatrix} + \tilde{\boldsymbol{C}} \begin{bmatrix} \dot{\boldsymbol{r}}_0 \\ \bar{\boldsymbol{u}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \tilde{\bar{\boldsymbol{M}}}_{\mathcal{I}} + \nabla \bar{\boldsymbol{M}}_{\mathcal{I}} \left(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}} \right) \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \bar{\boldsymbol{M}} \end{bmatrix} \tag{49}$$

Figure 8 shows the results of the second optimization step. During the optimization, the maneuver cost was reduced from 37.6 N^2 m^2 s of the initial guess down to 8.9 N^2 m^2 s for the converged solution.

CONCLUSION

Obtaining an algorithm capable to optimize, on-board and in real-time, the maneuvers of a spacecraft-manipulator systems is challenging. With its computational advantages, convex optimization seems a potentially suitable approach. In this paper, we showed that the optimization can be simplified by dividing the maneuver into two sub-maneuvers (system-wide translation and internal re-configuration) and individually optimizing these two sub-maneuvers. By splitting the maneu-

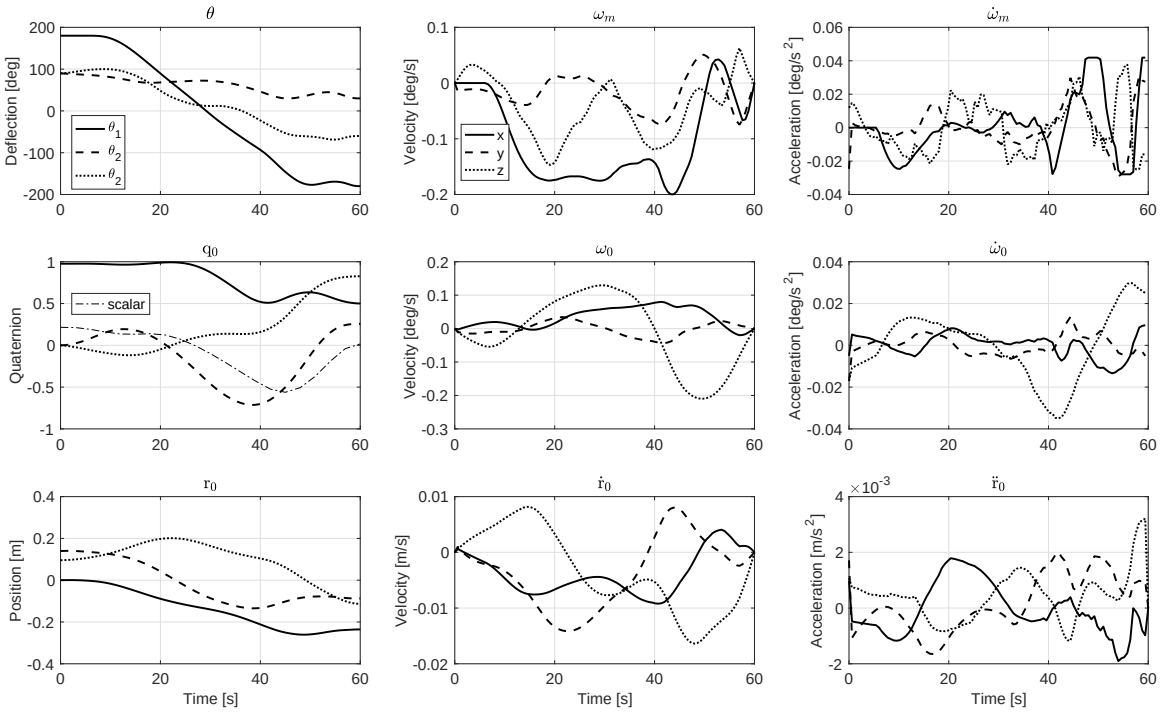


Figure 8. Internal Re-Configuration Maneuver Results.

ver, we are required to make a conservative assumption on the size of the chaser system to ensure a collision-free motion, resulting in a sub-optimal solution. The maneuver division also forces us to prioritize the translation cost minimization, even at the expense of the internal re-configuration cost. The presence of non-convex state inequality constraints and nonlinear dynamics is circumvented by using a sequential convex optimization approach. The resulting approach obtains a local solution to the original optimal control problem by solving a collection of simpler convex programming problems. An explicitly convex formulation of a line-of-sight constraint has also been derived.

REFERENCES

[1] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Progress in Aerospace Sciences*, Vol. 68, 2014, pp. 1 – 26, 10.1016/j.paerosci.2014.03.002.

[2] O. P. Agrawal and Y. Xu, "On the global optimum path planning for redundant space manipulators," *IEEE transactions on systems, man, and cybernetics*, Vol. 24, No. 9, 1994, pp. 1306–1316.

[3] R. Lampariello, S. Agrawal, and G. Hirzinger, "Optimal motion planning for free-flying robots," *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, Vol. 3, Sept 2003, pp. 3029–3035 vol.3, 10.1109/ROBOT.2003.1242056.

[4] F. Aghili, "Optimal control for robotic capturing and passivation of a tumbling satellite with unknown dynamics," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Vol. 21, 2008, pp. 2008–7274, 10.2514/6.2008-7274.

[5] T. Oki, H. Nakanishi, and K. Yoshida, "Time-optimal manipulator control of a free-floating space robot with constraint on reaction torque," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2008, pp. 2828–2833.

[6] A. Flores-Abad, Z. Wei, O. Ma, and K. Pham, "Optimal Control of Space Robots for Capturing a Tumbling Object with Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 37, 2015/08/27 2014, pp. 2014–2017, 10.2514/1.G000003.

[7] A. Flores-Abad, L. Zhang, Z. Wei, and O. Ma, "Optimal Capture of a Tumbling Object in Orbit Using a Space Manipulator," *Journal of Intelligent & Robotic Systems*, 2016, pp. 1–13, 10.1007/s10846-016-0417-1.

[8] H. Hanßmann, *Hamiltonian Systems with Three or More Degrees of Freedom*, ch. Quasi-Periodic Motion of a Rigid Body Under Weak Forces, pp. 398–402. Springer Netherlands, 1999.

[9] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization.," *Robotics: science and systems*, Vol. 9, Citeseer, 2013, pp. 1–10.

[10] X. Liu and P. Lu, "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765, 10.2514/6.2013-4725.

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] D. P. Scharf, B. Açkmeşe, D. Dueri, J. Benito, and J. Casoliva, "Implementation and Experimental Demonstration of Onboard Powered-Descent Guidance," *Journal of Guidance, Control, and Dynamics*, 2017/02/14 2016, pp. 1–17, 10.2514/1.G000399.

[13] D. Dueri, B. Açkmeşe, D. P. Scharf, and M. W. Harris, "Customized Real-Time Interior-Point Methods for Onboard Powered-Descent Guidance," *Journal of Guidance, Control, and Dynamics*, 2017/02/14 2016, pp. 1–16, 10.2514/1.G001480.

[14] B. Açıkmeşe, J. M. Carson, and L. Blackmore, "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem," *IEEE Transactions on Control Systems Technology*, Vol. 21, No. 6, 2013, pp. 2104–2113.

[15] P. Lu and X. Liu, "Autonomous trajectory planning for rendezvous and proximity operations by conic optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 375–389.

[16] Y. Umetani and K. Yoshida, "Workspace and manipulability analysis of space manipulator," *Trans. Soc. Instrum. Control Eng*, Vol. E-1, No. 1, 2001, pp. 116–123.

[17] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal on Robotics and Automation*, Vol. 4, No. 2, 1988, pp. 193–203.

[18] G. v. d. Bergen, "A fast and robust GJK implementation for collision detection of convex objects," *Journal of graphics tools*, Vol. 4, No. 2, 1999, pp. 7–25.

[19] R. Pinson and P. Lu, "Trajectory Design Employing Convex Optimization for Landing on Irregularly Shaped Asteroids," *AIAA/AAS Astrodynamics Specialist Conference, Long Beach, California 13 - 16 September 2016.*, American Institute of Aeronautics and Astronautics, 2016, doi:10.2514/6.2016-5378.

[20] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Swarm Assignment and Trajectory Optimization Using Variable-Swarm, Distributed Auction Assignment and Sequential Convex Programming," *The International Journal of Robotics Research*, 2016.

[21] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control*. Springer, 2009.

[22] Y. Kim, M. Mesbahi, G. Singh, and F. Y. Hadaegh, "On the Convex Parameterization of Constrained Spacecraft Reorientation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 46, July 2010, pp. 1097–1109, 10.1109/TAES.2010.5545176.

[23] U. Lee and M. Mesbahi, "Constrained Autonomous Precision Landing via Dual Quaternions and Model Predictive Control," *Journal of Guidance, Control, and Dynamics*, 2017/01/23 2016, pp. 0–0, 10.2514/1.G001879.

[24] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," `http://cvxr.com/cvx`, Mar. 2014.

[25] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," *Recent Advances in Learning and Control* (V. Blondel, S. Boyd, and H. Kimura, eds.), Lecture Notes in Control and Information Sciences, pp. 95–110, Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[26] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3—a MATLAB software package for semidefinite programming, version 1.3," *Optimization methods and software*, Vol. 11, No. 1-4, 1999, pp. 545–581.

[27] J. Virgili-Llop, D. V. Drew, and M. Romano, "Spacecraft Robotics Toolkit: An Open-Source Simulator for Spacecraft Robotic Arm Dynamic Modeling and Control.," *6th International Conference on Astrodynamics Tools and Techniques*, 2016.

[28] J. Virgili-Llop *et al.*, "SPART: SPAcecraft Robotics Toolkit," `https://github.com/NPS-SRL/SPART`.